11 | 性能测试的工具：七大测试场景如何选择高质量的测试工具？

2019-12-20 庄振运

性能工程高手课 进入课程 >

讲述：庄振运
时长 12:28　大小 11.42M

你好，我是庄振运。

我们在前面两讲讨论了如何进行性能测试的规划和设计。性能测试离不开合适的工具，那么这一讲，我们来讨论一下测试工具的分类和构成，并根据七个不同的测试场景，分别学习几个常用的高质量测试工具，尤其是开源的。

这七个测试场景分别是：Web 测试、系统测试、数据库测试、文件 IO 测试、存储测试、网络测试以及移动 App 测试。

我们应该尽量借助这些好用的工具，而不要自己去重新开发。

自己开发的话，费时费力不说，开发出来的工具也不见得会比这些工具好用。更重要的是，性能测试工具的开发和使用过程中，有很多需要注意的地方和容易陷入的坑，一不小心就会掉坑。我们下一讲就会讨论常见的坑。

而这些常用的工具已经经过很多人的努力和长时间的改进，在很多方面避免了容易出现的各种问题，所以尽量使用它们吧。

测试工具的分类

首先你必须认识到，性能测试工具是繁多的。之所以繁多，是因为每种工具适合的场合不同，所以它们各有特点。比如如下几个方面：

测试场景：是针对 Web 环境、移动 App、系统、数据库，还是模块测试？

测试类型：是基准测试还是峰值测试？

免费还是收费：开源工具一般都是免费的；但是很多收费工具也的确物有所值。

支持的协议：比如是否支持 HTTP 协议、FTP 协议等等。

支持的功能：比如并发性支持度，能否分析测试结果，能否录制性能测试脚本等。

测试工具的模块

要评价一个测试工具的优劣好坏，我们就需要知道测试工具的模块和测试的一般过程。

大规模性能测试的一般过程是：通过录制、回放定制的脚本，模拟多用户同时访问被测试系统（SUT）来产生负载压力，同时监控并记录各种性能指标，最后生成性能分析结果和报告，从而完成性能测试的基本任务。

比照这个过程，一个稍微健全的测试工具都会包括以下的模块：

负载生成模块：负责产生足够的流量负载。多少流量算足够？这得根据测试类型和具体需求来定。如果测试类型是压力测试，那么产生的流量一定要大到 SUT 不能处理的程度。

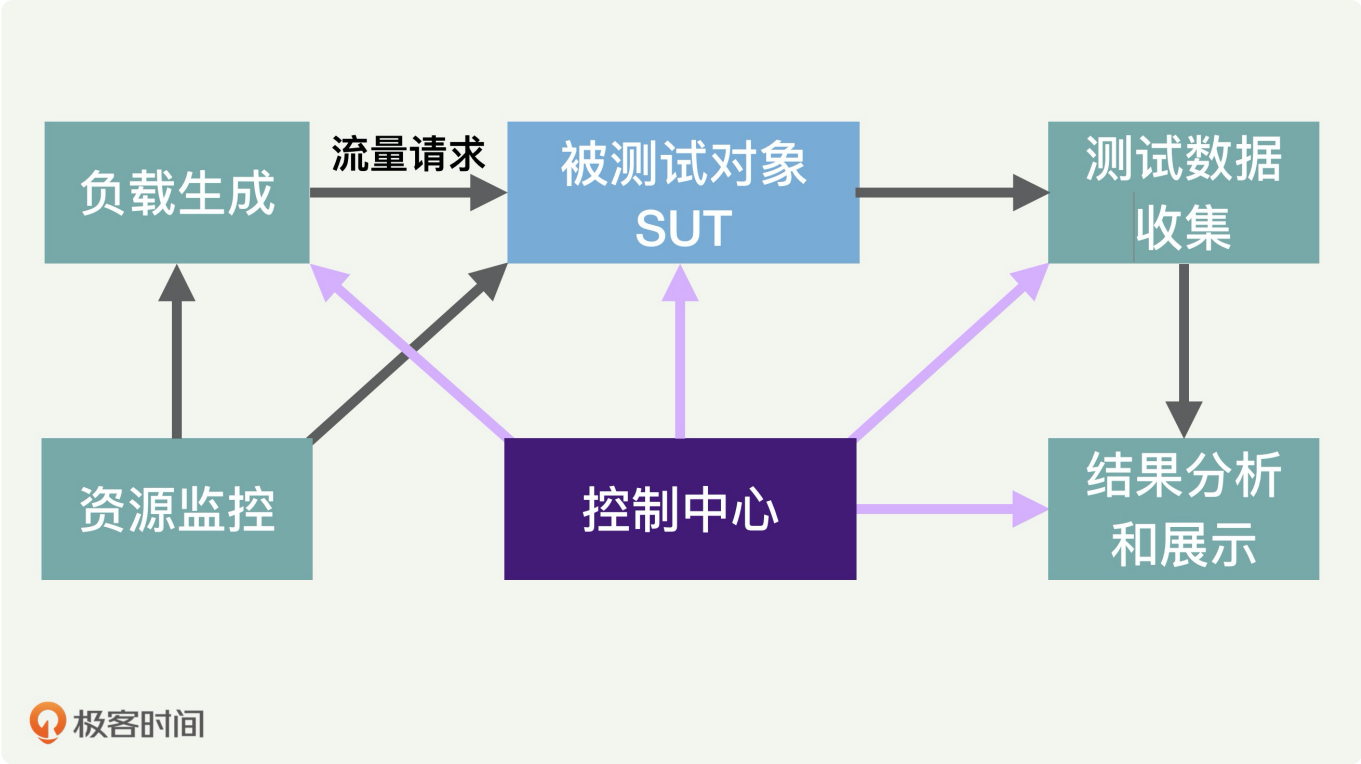
测试数据收集模块：负责获取测试的数据，包括具体的各种性能数据。这个收集可以是实时的，就是在测试进行之中收集；也可以是后期，等测试完成之后收集的。

结果分析和展示：有了大量的测试数据，就需要进行分析并展示。同样的，这个过程可以是实时的，也可以是等全部测试完成后进行。

资源监控模块：测试过程中离不开对 SUT 和流量生成模块的实时资源监控，目的是确保这两个模块运行正常。具体来说，流量生成模块必须不能负载过量，否则很可能产生的流量不够大。SUT 也要确保运行还是正常的，否则整个测试就失去意义了。

控制中心模块：测试者需要用这个模块和整个测试系统来交互，比如开始或停止测试，改变测试的各种参数等等。

这几个模块的关系我用下面这张图表示。



Web 测试场景

我们先看第一个测试场景：Web 测试。这一场景的测试工具很多，我们介绍几个。

JMeter 是一款优秀而小巧精致的开源测试工具，是用 Java 写的。JMeter 安装简单，使用方便，所以很流行，建议每个性能测试者都掌握它。熟练使用它，在绝大多数场合都能大大提高测试效率。JMeter 的测试是基于 HTTP 协议的，所以最好对 HTTP 协议熟悉一些才能快速上手和理解里面的概念。

LoadRunner 是 HP 公司的一款测试工具，功能和资料都比较全，也好用，但不是开源的。它的组成模块都很强大，比如分析模块中的 AutoCorrelation 向导。这个向导会自动整理所有的监控和诊断数据，并找出导致性能降低的最主要的几个原因。这样就将性能测试结果转化为可处理的精确数据，从而使开发团队大大减少了解决问题的时间。

Locust 是基于 Python 的开源测试工具，支持 http、https 等协议。它的一个突出优点是可扩展性很好。

系统测试场景

这种场景下的测试工具很多，我主要介绍两个。

UnixBench 是一个 Unix 系统（比如 Unix、BSD、Linux）下的性能测试工具，是开源的，而且被普遍用于测试 linux 系统主机的性能。这个工具可以测试很多模块和场景，比如系统调用、读写、进程、图形化测试、2D、3D、管道、运算、C 库等，它的测试结果可以作为基准性能测试数据。

比如，你可以用它测试从一个文件向另外一个文件传输数据的速率，要求每次测试使用不同大小的缓冲区。再比如测试两个进程通过一个管道（Pipe）交换一个不断增大的整数的速度；类似现实编程中的一些应用，这个测试会首先创建一个子进程，再和这个子进程进行双向的管道传输。它也可以测试进入和离开操作系统内核的开销，即一次系统调用（System Call）的开销代价。

这是通过反复地调用 getpid 函数的小程序来进行的。

Perf 是 Linux 下最普遍使用的性能分析工具，功能强大全面，俗称性能测试的“瑞士军刀”。比如，Perf 可以对程序进行函数级别的采样，从而了解程序的性能瓶颈究竟在哪里。或者计算每个时钟周期内的指令数等等。

Perf 的原理是使用特殊的计数器来进行性能统计。它既可以分析指定应用程序的性能问题，也可以用来分析内核的性能问题，所以可以全面理解应用程序中的性能瓶颈。

我个人建议每个关心性能的人都了解和学习一下 Perf 的使用。

数据库测试场景

数据库的测试工具也是汗牛充栋。

SysBench 是一个容易使用的的开源多线程测试工具，主要用于测试数据库性能，比如 MySQL， Oracle 和 PostgreSQL，但也可以测试 CPU，内存，文件系统等性能。它的强项包括数据分析和展示模块，多线程并发性比较好，而且开销低。另外我们可以很容易地定制脚本，来创建新的测试。

mysqlslap 是 Mysql 自带的压力测试工具，它可以轻松模拟出大量客户端同时操作数据库的情况。

文件 IO 和存储测试场景

对文件系统和存储系统的性能测试工具也有很多，比如 **ioZone**，可以测试不同操作系统中的文件系统的读写性能。比如可以测试不同 IO 读写方式下硬盘的性能。

Bonnie++ 是一个用来测试 UNIX 文件系统和磁盘性能的测试工具，它可以通过一系列的简单测试来生成硬盘和文件系统的性能参数。这个工具很容易使用，输出结果显示方面很不错。

如果你希望在 Linux 中很快地测试硬盘读写性能，**dd** 这个很有用的命令经常就够用了。这个工具就是用指定大小的块拷贝一个文件，并在拷贝的同时进行指定的转换。

网络测试场景

网络测试也有各种各样的工具。

Netperf 是很不错的一个网络性能的测量工具，主要针对基于 TCP 或 UDP 的传输。它有两种基本模式，即批量数据传输（bulk data transfer）模式和请求 / 应答（request/reponse）模式，我们可以根据应用的不同来选择不同模式。

测试结果所反映的，是两个系统之间发送和接受数据的速度和效率，即一个系统能够以多快的速度向另一个系统发送数据，以及后者能够以多快的速度接收数据。

Iperf 可以测试最大 TCP 和 UDP 带宽性能，具有多种参数和 UDP 特性，可以根据需要调整，可以报告带宽、延迟抖动和数据包丢失。这个工具使用起来很简单，一条命令行就可

以。比如如下的测试：“采用 UDP 协议，以 100Mbps 的数据发送速率，从本机到服务器 192.168.1.1 作上传带宽测试；测试时间为 120 秒“，或者“以 50Mbps 的发送速率，同时向服务器端发起 100 个连接线程 “。

移动 App 测试场景

移动 App 测试的性能指标主要是内存、CPU、电量使用、启动时长、显示帧率、网络流量等。 针对 App 的性能测试工具和平台可以按照两种方式分类。

第一种分类是根据**线下还是线上**。线下 App 性能测试主要依靠传统测试手段和方法，比如不同的版本，框架等等都需要进行 App 性能测试。 线上测试算是场景化测试，主要针对大规模或者动态环境，让 App 在特定场合和特别条件下，更精准地衡量 App 的核心性能。

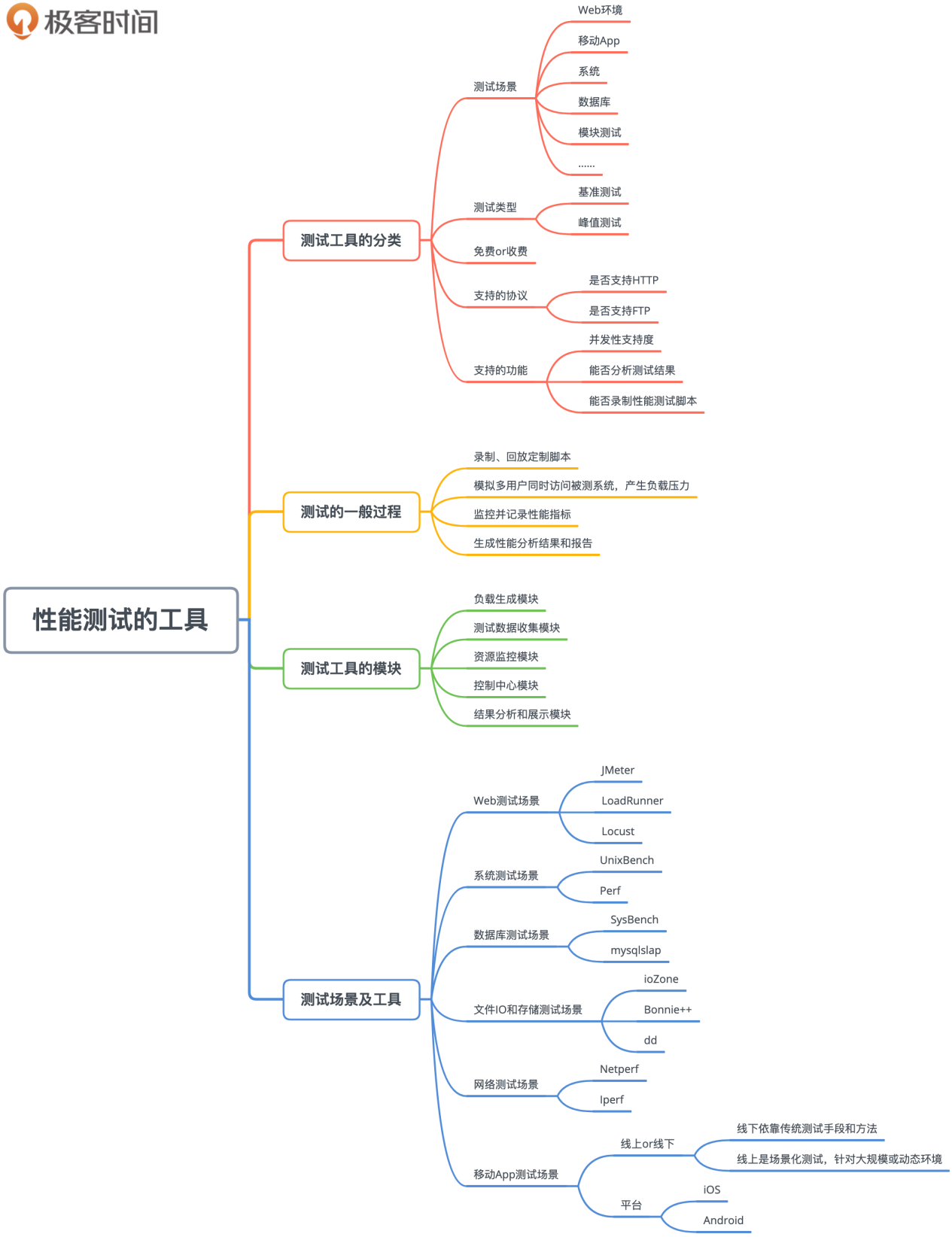
另外一种分类方式是**平台生态系统**，现在主要有安卓 Android 和 iOS。有的测试工具可以兼容多个平台，比如 **Appium**，就是一个可以同时支持安卓和 iOS 的测试框架的工具，功能强大。

对 Android 的测试工具常用的有 **adb**（Android Debug Bridge）和 **Monkey**。Monkey 是 Android SDK 自带的测试工具。Monkey 的意思是猴子， 顾名思义，就是在电脑面前乱敲键盘在测试。 在测试过程中会向系统发送伪随机的用户事件流，如按键输入、触摸屏输入、手势输入等，对 App 进行压力测试，也有日志输出。

对 iOS App 的测试工具也很多，比如 XCTest、Frank、KIF 等等，这里就不展开了。

总结

古人说：“工欲善其事，必先利其器”。业界已经有很多好用的工具，一般都能满足大多数场合的测试要求，会让我们的测试工作如虎添翼。



虽然有些环境下我们经常倾向于自己开发某些工具，但是往往会花费很多时间，而且很容易陷入各种各样的坑中，因此尽量避免自己开发。

这一讲介绍的几款工具适用于不同的场合，如果你能合理运用它们，应该会取得事半功倍的效果。

思考题

你曾经用过什么样的性能测试工具？

为什么会选择这些工具，而不是其他的工具？

你使用的测试工具有什么优点和缺点？

你是怎么克服它的缺点的？

欢迎你在留言区分享自己的思考，与我和其他同学一起讨论，也欢迎你把文章分享给自己的朋友。

© 版权归极客邦科技所有，未经许可不得传播售卖。 页面已增加防盗追踪，如有侵权极客邦将依法追究其法律责任。

上一篇　10 | 性能测试的规划和步骤：为什么性能测试不容易一蹴而就呢？

下一篇　12 | 九条性能测试的经验和教训：如何保证测试结果可靠且可重复？

精选留言 (1)

写留言

志鑫

2019-12-21

常用wrk，支持lua脚本

展开